**main.dart**

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'task\_viewModel.dart';

import 'home\_view.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return ChangeNotifierProvider(

      create: (context) => TaskViewModel(),

      child: MaterialApp(

        debugShowCheckedModeBanner: false,

        title: 'To-Do App',

        theme: ThemeData(

          primarySwatch: Colors.blue,

        ),

        home: HomeView(),

      ),

    );

  }

}

**home\_view.dart**

(view component)

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'task\_viewModel.dart';

import 'task\_item.dart';

class HomeView extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    final taskViewModel = Provider.of<TaskViewModel>(context);

    final titleController = TextEditingController();

    final descriptionController = TextEditingController();

    void \_addTask() {

      final title = titleController.text.trim();

      final description = descriptionController.text.trim();

      if (title.isNotEmpty && description.isNotEmpty) {

        taskViewModel.addTask(title, description);

        titleController.clear();

        descriptionController.clear();

      } else {

        ScaffoldMessenger.of(context).showSnackBar(

          const SnackBar(

            content: Text("Both title and description are required!"),

          ),

        );

      }

    }

    return Scaffold(

      appBar: AppBar(

        title: const Text('To-Do App'),

        backgroundColor: Colors.blueAccent,

      ),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          children: [

            // Text fields for adding tasks

            TextField(

              controller: titleController,

              decoration: const InputDecoration(

                labelText: 'Title',

                border: OutlineInputBorder(),

              ),

            ),

            const SizedBox(height: 10),

            TextField(

              controller: descriptionController,

              decoration: const InputDecoration(

                labelText: 'Description',

                border: OutlineInputBorder(),

              ),

            ),

            const SizedBox(height: 10),

            ElevatedButton(

              onPressed: \_addTask,

              child: const Text('Add Task'),

            ),

            const SizedBox(height: 20),

            // Display the list of tasks

            Expanded(

              child: ListView.builder(

                itemCount: taskViewModel.tasks.length,

                itemBuilder: (context, index) {

                  final task = taskViewModel.tasks[index];

                  return TaskItem(task: task);

                },

              ),

            ),

          ],

        ),

      ),

    );

  }

}

**task\_viewModel.dart**

(view model component)

import 'package:flutter/material.dart';

import 'task.dart';

class TaskViewModel extends ChangeNotifier {

  // List to store tasks

  List<Task> \_tasks = [];

  // Getter to access tasks

  List<Task> get tasks => \_tasks;

  // Method to add a task

  void addTask(String title, String description) {

    final newTask = Task(title: title, description: description);

    \_tasks.add(newTask);

    notifyListeners();

  }

}

**task.dart**

(model component)

class Task {

  final String title;

  final String description;

  // Constructor for Task

  Task({required this.title, required this.description}); }

**task\_item.dart**

extra component breaking down the complexity, can be thought of as a view component as it displays individual tasks

import 'package:flutter/material.dart';

import 'task.dart';

class TaskItem extends StatelessWidget {

  final Task task;

  const TaskItem({Key? key, required this.task}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    return Card(

      margin: const EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),

      elevation: 4,

      shape: RoundedRectangleBorder(

        borderRadius: BorderRadius.circular(8.0),

      ),

      child: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.start,

          children: [

            Text(

              task.title,

              style: const TextStyle(

                fontSize: 18,

                fontWeight: FontWeight.bold,

              ),

            ),

            const SizedBox(height: 8),

            Text(

              task.description,

              style: const TextStyle(

                fontSize: 16,

                color: Colors.grey,

              ),

            ),

          ],

        ),

      ),

    );

  }

}

Changes made in pubspec.yaml:

We added this under dependencies:

provider: ^6.1.2

So now it becomes:  
 dependencies:

| flutter:

| | sdk: flutter

| provider: ^6.1.2